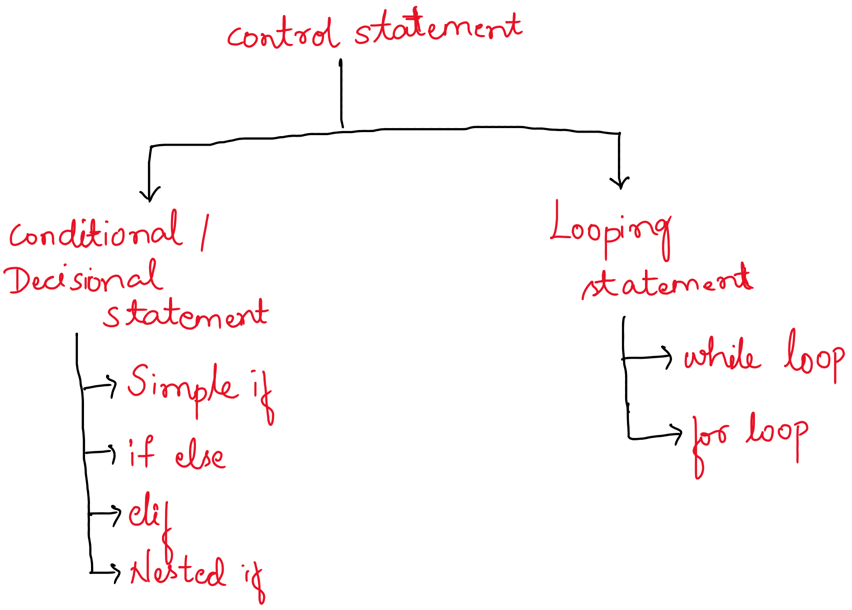
20 February 2025 18:46

Day-16

Control Statement:

--- It is used to control the flow of execution.

Types:



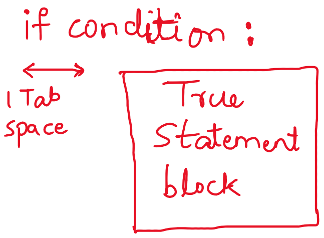
Conditional Statement:

--- It is used to control the flow of execution based on conditions.

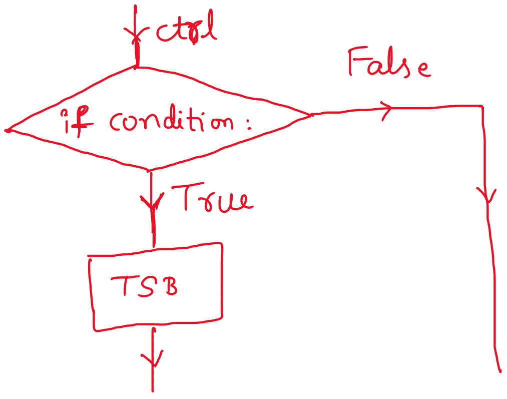
1) Simple if:

--- It is a keyword which is used to check the condition and it will execute the statement block if the condition is True or else it will ignore the statement block.

Syntax:



Flow diagram:



Programs:

```
# Simple if

# WAP to check whether the number is even.
'''
n = int(input('Enter the number: '))
if  n%2 == 0:
    print('number is even') '''

# WAP to check whether the string has exactly 5 characters in it.
'''
s = input('Enter the string: ')
if len(s)==5:
    print('string has exactly 5 characters in it')'''

# WAP to check whether the number is greater than 200.
'''
n = int(input('Enter the number: '))
if  n>200:
    print('number is greater than 200')'''

# WAP to print the square of the number only if it is multiple of 3.
'''
n = int(input('Enter the number: '))
if  n%3==0:
    print('square of the number is: ',n**2)'''

# WAP to check whether the number is 2 digit number.
'''
n = int(input('Enter the number: '))
if  n>=10 and n<=99:
    print('number is 2 digit number')'''

# WAP to check if the character is Uppercase.
'''
ch = input('Enter a character: ')
if  'A'<= ch <= 'Z':
    print('character is Uppercase')'''
```

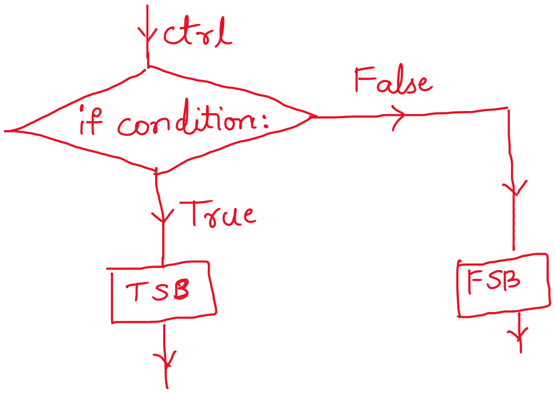
2) if else:

--- It is used to check the condition and it will execute the True Statement block if the condition is True else it will execute the False Statement block.

Syntax:

```
if condition :
⟷   TSB
else:
    FSB
```

Flow diagram:



Programs:

```
# if else

# WAP to check the given data is float or not.
```

```
'''
data =  eval(input('Enter the data: '))
if  type(data)==float:
   print('given data is float')
else:
   print('given data is not float')'''


# WAP to check whether the string is palindrome or not.
'''
s = input('Enter the string: ')
if s==s[::-1]:
   print('string is palindrome')
else:
   print('string is not palindrome')'''

# WAP to check whether the given character is vowel or not.
'''
ch = input('Enter the character: ')
if ch in 'aeiouAEIOU':
   print('given character is vowel')
else:
   print('given character is not vowel')'''


# WAP to check whether the given data is SVDT or not.
'''
data = eval(input('Enter the data: '))
if  type(data) in [int, float, complex, bool]:
   print('given data is SVDT')
else:
   print('given data is not SVDT')'''


# WAP to check whether the given integer is 3 digit number or not.
'''
n = abs(int(input('Enter the number: ')))
if  100<=n<=999:
   print('given integer is 3 digit number')
else:
   print('given integer is not 3 digit number')'''
```

Note:

abs (absolute function) - It will convert the negative numbers into positive numbers. If we already have positive number it will keep as it is.

Day-17

3) elif:

--- Whenever we want to check the multiple conditions and to execute statement blocks of each and every condition we use elif.

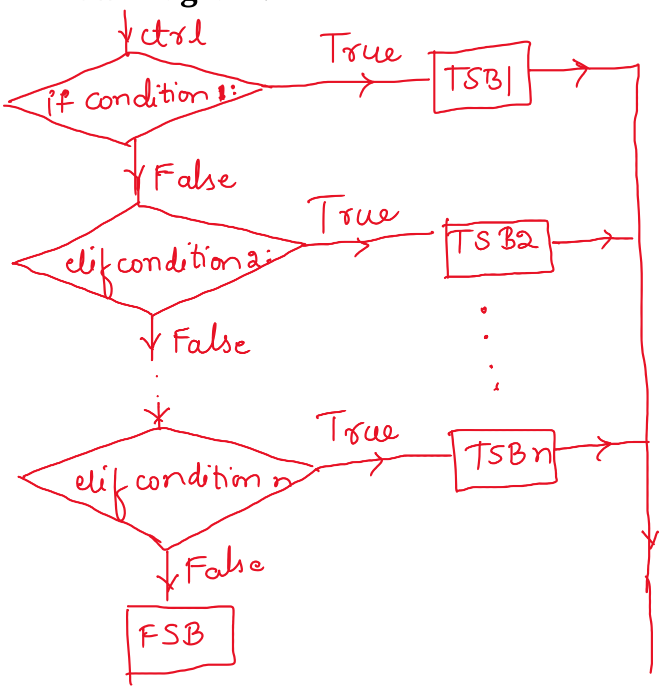
Syntax:

Flow Diagram:

Syntax:

```
if condition1 :
    TSB1 .
elif condition2 :
    TSB2 .
.
.
.
elif condition n :
    TSBn
else:
    FSB
```

Flow Diagram:



Programs:

```
# elif

# WAP to find the relation between 2 numbers.
'''
a = int(input('Enter the number1: '))
b = int(input('Enter the number2: '))
if a > b:
    print(a, 'is greater')
elif a < b:
    print(a, 'is lesser')
else:
    print(a, b,'are equal')'''

# WAP to check whether the character is uppercase or lowercase or digits or special characters
'''
ch = input('Enter the character: ')
if 'A'<=ch<='Z':
    print('character is uppercase')
elif 'a'<=ch<='z':
    print('character is lowercase')
elif '0'<=ch<='9':
    print('character is digit')
else:
    print('character is special character')'''

# WAP to check whether the number is single digit or  two digit or three digit or more than 3 digit.
'''
n = abs(int(input('Enter the number: ')))
if 0<=n<=9:
    print('single digit')
elif 10<=n<=99:
    print('two digit')
elif 100<=n<=999:
    print('three digit')
else:
    print('more than three digit')'''
```

```
# WAP to find the greatest among four numbers
'''
a = int(input('Enter the number1: '))
b = int(input('Enter the number2: '))
c = int(input('Enter the number3: '))
d = int(input('Enter the number4: '))
if a>b and a>c and a>d:
    print(a,'is greatest')
elif b>a and b>c and b>d:
    print(b,'is greatest')
elif c>a and c>b and c>d:
    print(c,'is greatest')
else:
    print(d,'is greatest')'''

# Assignment:  WAP to find the smallest among four numbers

# WAP to predict the student result based on the obtained percentage.
'''
per = float(input('Enter the percentage: '))
if per < 0 or per>100:
    print('Invalid result')
elif 70<=per<=100:
    print('Distinction')
elif 60<=per<70:
    print('First Class')
elif 45<=per<60:
    print('Second Class')
elif 35<=per<45:
    print('Just pass')
elif per<35:
    print('Fail')'''
```

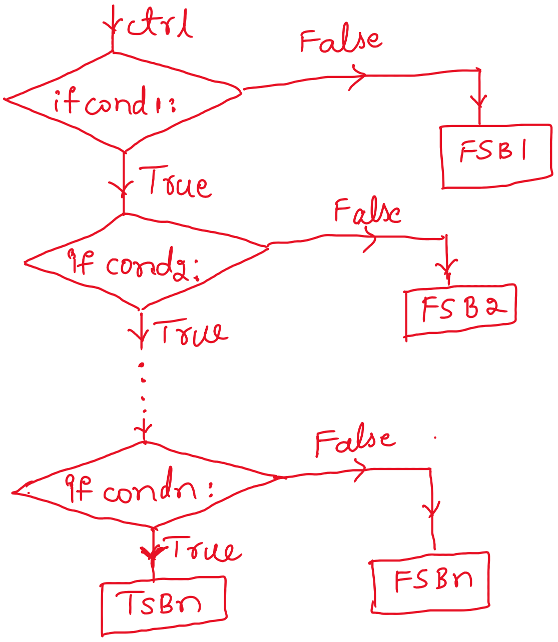
4) Nested if:

--- Whenever it is necessary to check a condition before checking another condition we use Nested if.

Syntax:

```
if cond1 :
    if cond2 :
          . . .
              if condn :
                  TSBn
              else:
                  FSBn
    else :
        FSB2
else :
    FSB1
```

Flow diagram:



Programs:

```
# Nested if

# WAP to check whether the given character is vowel or consonant.
'''
s = input('Enter the character: ')
if 'A'<=s<='Z' or 'a'<=s<='z':
    if s in 'aeiouAEIOU':
        print('Vowels')
    else:
        print('Consonants')
else:
    print('character is not alphabet')'''

# WAP to login to Instagram by entering the proper username and password.
'''
username = 'python'
password = 'coders@123'
un = input('Enter the username: ')
pw = input('Enter the password: ')
if un == username:
    if pw == password:
        print('Login Successful')
    else:
        print('Invalid password')
else:
    print('Incorrect username')'''

# WAP to print the greatest among 3 numbers
'''
a = int(input('Enter the number1: '))
b = int(input('Enter the number2: '))
c = int(input('Enter the number3: '))
if a>b:
    if a>c:
        print(a,'is greatest')
    else:
        print(c,'is greatest')
elif b>a:
    if b>c:
        print(b,'is greatest')
    else:
        print(c,'is greatest')'''
```

Day-18

Looping Statement:

--- It is a control statement which will control the flow of execution by repeating the same task again and again.

Types:

- While loop
- For loop

1) While loop:

--- It is used to execute the same set of instructions again and again until the condition become False.

Note:

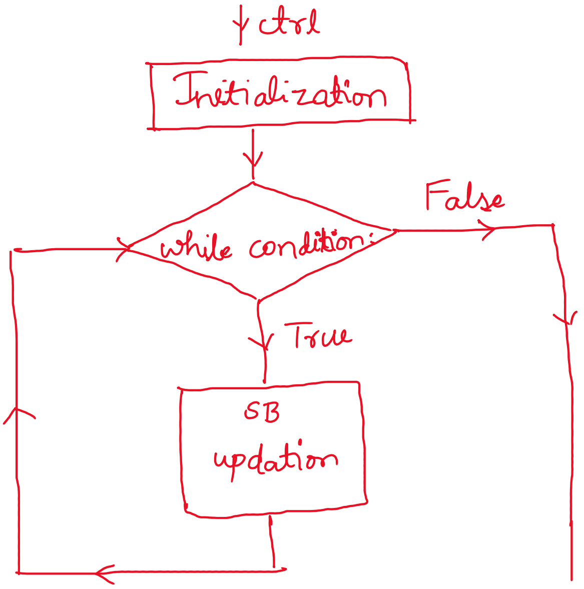
○ **Initialization**

○ **Updation**

Syntax:

```
initialisation
while condition :
    SB
    updation
```

Flow Diagram:



Programs:

```
# while loop

# WAP to print hello world for 5 times
'''
i = 1
while i<=5:
    print('hello world')
    i = i + 1'''
```

Tracing

i	i<=5 :	print('hello world')	i=i+1

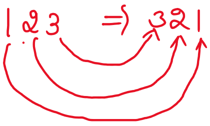
i=1	1<=5 :✓	hello world	i=1+1
			i=2

i=2	2<=5 :✓	hello world	i=2+1
			i=3

i=3	3<=5 :✓	hello world	i=3+1 => 4
i=4	4<=5 :✓	hello world	i=4+1 => 5
i=5	5<=5 :✓	hello world	i=5+1 => 6
i=6	6<=5 : X		

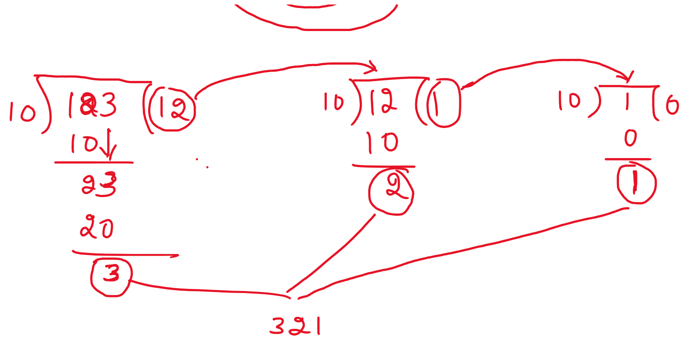
```
# WAP to print first 10 natural numbers.
'''
num = int(input('Enter the number: '))
i = 1
while i <= num:
    print(i)
    i = i + 1 '''
```

```
# WAP to print the  first 10 natural numbers in reverse order
'''
n = int(input('Enter the number: '))
i = n
while i >0:
    print(i)
    i = i - 1'''
```



```
   print(i)
   i = i - 1'''

# WAP to print all the even numbers from 1 to 50
'''
n = int(input('Enter the number: '))
i = 1
while i <= 50:
   if i%2==0:
      print(i)
   i = i + 1'''
```



```
'''
n = int(input('Enter the number: '))
i = 2
while i <= 50:
   print(i)
   i = i + 2'''

# WAP to print the sum of n natural numbers
'''
n = int(input('Enter the number: '))
i = 1
res = 0
while i<=n:
   res = res + i
   i = i + 1
   print(res)'''
```

rev=0 ✓

```
# WAP to reverse the number without using typecasting.
'''
n = int(input('Enter the number: '))
rev = 0
while n > 0:
   rem = n % 10
   rev = rev*10 + rem
   n = n // 10
print(rev)'''
```

		rem	rev	
n	n>0	n % 10	rev*10 +rem	n = n // 10
n = 123	123>0:✓	123%10 ⇒3	⇒ 0*10+3 rev = 3	= 123 // 10 n = 12
n= 12	12>0:✓	12%10 ⇒2	⇒3*10 +2 rev=32	=12 //10 n =1
n=1	1>0:✓	1 % 10 ⇒ 1	⇒ 32*10+1 rev ⇒321	=1 //10 = 0
n=0	0>0:✗			

```
# WAP to print the product of individual digit
from the number.
'''
n = int(input('Enter the number: '))
prod = 1
while n > 0:
   rem = n%10
   prod = prod * rem
   n = n // 10
print(prod)'''
```

143

1*4*3 ⇒ 12

3*4*1 ⇒ 12

Assignment: Do tracing for this program